On Computer Algebra Generation of Symplectic Integrator Methods

or

Of Headaches, Nightmares, and Algebra

Marc A. Murison (USNO)
and
John E. Chambers (Armagh Obs.)

30th Meeting of the AAS Division on Dynamical Astronomy
Estes Park, Colorado
April 30, 1999

Note: the full set of slides for this talk is located on the web at

http://aa.usno.navy.mil/murison/talks/

ON COMPUTER ALGEBRA GENERATION OF SYMPLECTIC INTEGRATOR METHODS

Marc A. Murison (USNO) and John E. Chambers (Armagh Obs.)
murison@aa.usno.navy.mil and jec@star.arm.ac.uk

Abstract

Most symplectic integrators used in solar-system dynamics are second-order in the time step τ. Typically, the Hamiltonian is divided into a Keplerian piece H_A and a smaller perturbative component H_B. We can take advantage of the disparity in relative magnitude of these components to define a second small parameter, call it $\varepsilon = \frac{|H_B|}{|H_A|} \ll 1$, and use this to obtain a "partially" higher-order method.

Adopting a Lie series approach, one can, for a given order-N method, examine the τ^{N+1}, τ^{N+2}, etc. error terms. Each of the $2^k - 2$ subterms of the coefficient of the τ^k error term has an associated factor of ε raised to a power ranging from linear to $k-1$. By including adjustable parameters in each evolution operator $exp(\tau\{\cdot,H_A\})$ or $exp(\tau\{\cdot,H_B\})$ in the trial method (composed of a combination of these operators) that approximates the true Hamiltonian evolution operator $exp(\tau\{\cdot,H_A + H_B\})$, one can in principle eliminate specified subterms in specified error terms. For example, a second-order method chosen to eliminate the τ^3 subterms linear in ε can, depending on the magnitude of ε, produce a quasi-third-order method. In practice this process boils down to generating then solving systems of nonlinear polynomial equations particular to the trial method.

A computer algebra program has been developed that automates the generation and solution of the equations that result from requesting a specified method of order N. This task is tedious due to the noncommutative algebra involved in the series expansions and subsequent algebraic manipulations, but computers are well-suited for handling such tedium. Once a method, or set of equivalent methods, has been found, the program then generates and solves a second set of equations for parameter solutions whereby subterms of specified powers in ε are eliminated for successive τ^{N+1}, τ^{N+2}, etc. terms in the overall error expression.

The project has, in these initial stages, been at least partially successful. Experiences and results to date will be presented.

Symplectic Integrator Micro-Tutorial

- Hamilton's equations

$$\frac{d\vec{q}}{dt} = \frac{\partial H}{\partial \vec{p}}, \qquad \frac{d\vec{p}}{dt} = -\frac{\partial H}{\partial \vec{q}}$$

- Poisson bracket

$$\{F, H\} \equiv \frac{\partial F}{\partial \vec{q}} \cdot \frac{\partial H}{\partial \vec{p}} - \frac{\partial F}{\partial \vec{p}} \cdot \frac{\partial H}{\partial \vec{q}} \qquad \longrightarrow \qquad \frac{d\,\cdot}{dt} = \{\cdot, H\}$$

- Equations of motion

$$\frac{d\vec{\xi}}{dt} = \{\vec{\xi}, H\} \qquad \vec{\xi} \equiv (\vec{q}, \vec{p})$$

- Formal Solution

$$\begin{aligned} \vec{\xi}(t) &= e^{\tau\{\cdot, H\}}\vec{\xi}(t-\tau) \\ &= \left(1 + \tau\{\cdot, H\} + \tfrac{\tau^2}{2}\{\cdot, H\}^2 + \ldots\right)\vec{\xi}(t-\tau) \end{aligned}$$

$$\tau \equiv t - t_0 \qquad \{\cdot, H\}^2 = \{\{\cdot, H\}, H\}$$

- Recast as a mapping or evolution operator:

$$S(\tau) \equiv e^{\tau\{\cdot, H\}}$$

$$\vec{\xi}(t) = S(\tau)\vec{\xi}(t-\tau)$$

The mapping "updates" the system to the next time step — the basis for a symplectic integration algorithm

Symplectic Integrator Micro-Tutorial (cont.)

- Split the Hamiltonian into two parts

$$H = H_A + H_B$$

also define operators $A \equiv \{\cdot, H_A\}$ $B \equiv \{\cdot, H_B\}$

  - e.g., a Keplerian part and a perturbative part
  - Then we can write

$$\begin{aligned} S(\tau) &= e^{\tau(A+B)} \\ &= 1 + \tau(A+B) \\ &\quad + \tfrac{1}{2}\tau^2(A^2 + AB + BA + B^2) + \ldots \end{aligned}$$

- Multiplication is noncommutative: $[A, B] \equiv AB - BA \neq 0$
  - Makes algebra more difficult
- Practical algorithm: take separate "A" and "B" steps

$$\begin{aligned} S_A(\tau)S_B(\tau) &= e^{\tau A}e^{\tau B} \\ &= 1 + \tau(A+B) \\ &\quad + \tfrac{1}{2}\tau^2(A^2 + 2AB + B^2) + \ldots \end{aligned}$$

$$S_A(\tau) \equiv e^{\tau A} \qquad S_B(\tau) \equiv e^{\tau B}$$

- Differs from the real Hamiltonian operator starting in the second-order term
  - Hence, two exponential operators gives us a first-order symplectic method

Symplectic Integrator Micro-Tutorial (cont.)

- Here's the trick: assemble a sequence of exponential operators $S_A(\alpha_k\tau)$, $S_B(\alpha_k\tau)$ and judiciously choose coefficients α_k to match the true evolution operator to a given order in the time step.
- Example: three exponentials yield second-order methods
  - The approximate Hamiltonian operator

$$
\begin{aligned}
\tilde{S}(\tau) &\equiv S_A(a\tau)S_B(b\tau)S_A(c\tau) \\
&= e^{a\tau A}e^{b\tau B}e^{c\tau A} \\
&= (1 + aA\tau + \tfrac{1}{2}a^2A^2\tau^2 + \ldots) \\
&\quad \cdot(1 + bB\tau + \tfrac{1}{2}b^2B^2\tau^2 + \ldots) \\
&\quad \cdot(1 + cA\tau + \tfrac{1}{2}c^2A^2\tau^2 + \ldots) \\
&= 1 + [bB + (a+c)A]\tau \\
&\quad + [\tfrac{1}{2}(a+c)^2A^2 + abAB \\
&\qquad + bcBA + \tfrac{1}{2}b^2B^2]\tau^2 \\
&\quad + \ldots
\end{aligned}
$$

  - Difference from the true Hamiltonian operator

$$
\begin{aligned}
\tilde{S}(\tau) - S(\tau) = &\ [(a-1+c)A + (b-1)B]\tau \\
&+ [\tfrac{1}{2}(a+1+c)(a-1+c)A^2 \\
&\quad + (ab - \tfrac{1}{2})AB + (bc - \tfrac{1}{2})BA \\
&\quad + \tfrac{1}{2}(b-1)(b+1)B^2]\tau^2 \\
&+ \ldots
\end{aligned}
$$

Symplectic Integrator Micro-Tutorial (cont.)

- Yields an overdetermined system of equations

$$
\begin{aligned}
b-1&=0\\
a-1+c&=0\\
(b-1)(b+1)&=0\\
2bc-1&=0\\
2ab-1&=0\\
(a+1+c)(a-1+c)&=0
\end{aligned}
$$

- Solution: $\{a=c=\frac{1}{2},\ b=1\}$
- The resulting method is second-order in the time step:

$$
\begin{aligned}
\tilde{S}(\tau) \;=\;& e^{\tau(A+B)}\\
&+\tau^3\left(\tfrac{1}{12}[B,B,A]-\tfrac{1}{24}[A,A,B]\right)\\
&+O(\tau^4)
\end{aligned}
$$

where

$$[A,A,B]\equiv[A,[A,B]]=A^2B-2ABA+BA^2$$

$$[B,B,A]\equiv[B,[B,A]]=B^2A-2BAB+AB^2$$

- This is the traditional symmetric second-order solution

Symplectic Integrator Micro-Tutorial (cont.)

- Using this approach, we can in principle construct approximate symplectic evolution mappings that match the "real" mapping to any given order in the time step
- Unfortunately, in practice this rapidly becomes very difficult
  - Number of equations to solve = $2^n - 2$
  - Complexity of individual equations grows rapidly with time step order n
  - polynomial order of equations goes as n

Two Useful Insights

1. We can adjust the parameters to optimize the error terms more to our liking
   - Make use of a second small parameter:

     $$H = H_A + \varepsilon H_B \qquad \varepsilon \ll 1$$

   - Add extra exponential operators
     - more parameters to play with
   - Selectively eliminate certain subterms in the time step error terms
   - For example, the traditional second-order evolution operator becomes

$$\begin{aligned}\tilde{S}(\tau) \; = \; & e^{\tau(A+B)} \\ & +\tau^3\left(\tfrac{1}{12}\varepsilon^2[B,B,A] - \tfrac{1}{24}\varepsilon[A,A,B]\right) \\ & +O(\varepsilon\tau^4)\end{aligned}$$

Remove this term and we have a quasi-fourth-order method

Two Useful Insights (continued)

2. Tedious and voluminous algebra: this is what computers are for!
   - General-purpose computer algebra systems (CAS)
     - Maple, Macsyma, Mathematica, Axiom, etc.
   - Symbolic programming languages enable
     - flexibility
     - algebraic sophistication
     - automation

Plan of Attack: a Two-Stage Process

1. Create a symplectic method, but include one or more additional exponential operators
   - Hard!
   - For example, a second-order method composed of more than three substeps
2. Solve for values of the extra parameter(s) that will eliminate the desired error subterms
   - Even harder!
   - Requires solving a second, usually nastier, set of polynomial equations
   - For example, in a second-order method, eliminate the subterms of the τ^3 error expression that are linear in the Hamiltonian operator *B*
     - If *B* is the small one, then the remaining dominant error terms go as $\varepsilon^2\tau^3$
     - This leaves us with an essentially fourth-order method(!)
   - Costs us extra exponential terms
   - There are cases where the extra cost is still significantly smaller than that of going to the full higher-order method

Symbolic Program SYMPLECTIC

- Implemented the Plan of Attack in the Maple symbolic algebra programming language
  - About 2500 lines of symbolic manipulation code
- Main parts:
  - symplectic method solver
    - specify
      - number of exponential operators
      - parameter list
      - time step order of method, n
      - number of time step error terms to calculate beyond n, so that we can play with them in the...
  - targeted subexpression eliminator
    - input
      - symplectic method (solution generated by first part)
      - method error expression (can be HUGE)
      - number of time step error terms beyond n in which to eliminate subterms that are linear in A (or B)
    - output
      - optimized solutions
      - the full corresponding solution errors

Symbolic Program SYMPLECTIC (continued)

- Subsystems:
  - polynomial equation set solver(!)
    - use the Maple general solver as kernel of algorithm attuned to our specific equation set form
    - in practice, employ *both* methods and eliminate duplicate solutions
  - noncommutative algebra procedures
    - series expansions
    - truncated series multiplication
    - transformations
    - factoring
  - plotting procedures
  - utility procedures
    - algebraic manipulators and expression simplifiers

Example of a [7,3] solution

notice the two free parameters

$$
\begin{aligned}
S_4 = \Bigg[\, a = -\frac{1}{3}\Bigg(\Bigg(\Bigg(&\frac{1}{6}\,\frac{(72c^3d^2 - 60c^2d^2 + 72c^3d^4 + 12dc^2 - 144c^3d^3 + 6cd + 72c^2d^3 - 6cd^2 - 1)\,Z1}{12c^2d^2 + 1 - 12dc^2} + \frac{1}{2}\big(\\
&-576c^3d^4 + 1 + 72c^2d^4 + 288c^3d^5 - 144c^4d^2 - 72c^2d^2 - 2592c^5d^5 - 864c^4d^6 - 864c^5d^3 \\
&+ 2592c^5d^4 + 1728c^4d^5 + 864c^5d^6 - 10cd + 192c^3d^2 + 18cd^2 - 1296c^4d^4 + 576d^3c^4 \\
&+ 24c^2d^3\big) \Big/ (12c^2d^2 + 1 - 12dc^2)\Bigg)(36c^2d^2 + 3 - 36dc^2)\Bigg) \Bigg/ \Bigg(\\
&(12c^2d^2 + 1 - 12dc^2)(-36dc^2 + 3 - 72c^2d^3 - 12cd + 108c^2d^2 + Z1) \\
&\left(\frac{1}{6}\,\frac{Z1}{12c^2d^2 + 1 - 12dc^2} - \frac{1}{2}\,\frac{-12c^2d^2 + 1 - 12dc^2 + 24c^2d^3 + 4cd}{12c^2d^2 + 1 - 12dc^2}\right)\Bigg)\Bigg), \\
b = -\frac{1}{6}\,&\frac{Z1}{12c^2d^2 + 1 - 12dc^2} + \frac{1}{2}\,\frac{12c^2d^2 + 1 - 12dc^2 + 4cd - 2d}{12c^2d^2 + 1 - 12dc^2},\ c = c,\ d = d,\ e = \frac{1}{3}\Bigg(\\
&\left(2\,\frac{cd(2c-1)\,Z1}{12c^2d^2 + 1 - 12dc^2} - \frac{(2c-1)(72c^3d^4 - 72c^3d^3 + 12c^2d^2 + 12dc^2 - 6cd^2 - 1)}{12c^2d^2 + 1 - 12dc^2}\right) \\
&(36c^2d^2 + 3 - 36dc^2)\Bigg) \Bigg/ \Bigg((12c^2d^2 + 1 - 12dc^2) \\
&(-36dc^2 + 3 - 72c^2d^3 - 12cd + 108c^2d^2 + Z1) \\
&\left(\frac{1}{6}\,\frac{Z1}{12c^2d^2 + 1 - 12dc^2} - \frac{1}{2}\,\frac{-12c^2d^2 + 1 - 12dc^2 + 24c^2d^3 + 4cd}{12c^2d^2 + 1 - 12dc^2}\right)\Bigg), \\
f = \frac{1}{6}\,&\frac{Z1}{12c^2d^2 + 1 - 12dc^2} - \frac{1}{2}\,\frac{12dc^2 - 1 + 24c^2d^3 + 4cd - 36c^2d^2}{12c^2d^2 + 1 - 12dc^2},\ g = -\frac{1}{3}\Bigg(\Bigg(\Bigg(\\
&\frac{1}{2}\,\frac{(2cd - 1)\,Z1}{12c^2d^2 + 1 - 12dc^2} \\
&- \frac{1}{2}\,\frac{-12cd^2 + 72c^3d^3 - 6cd - 72c^3d^2 + 108c^2d^2 + 1 - 12dc^2 - 72c^2d^3}{12c^2d^2 + 1 - 12dc^2}\Bigg) \\
&(36c^2d^2 + 3 - 36dc^2)\Bigg) \Big/ (-36dc^2 + 3 - 72c^2d^3 - 12cd + 108c^2d^2 + Z1)\Bigg)\,\Bigg]
\end{aligned}
$$

where

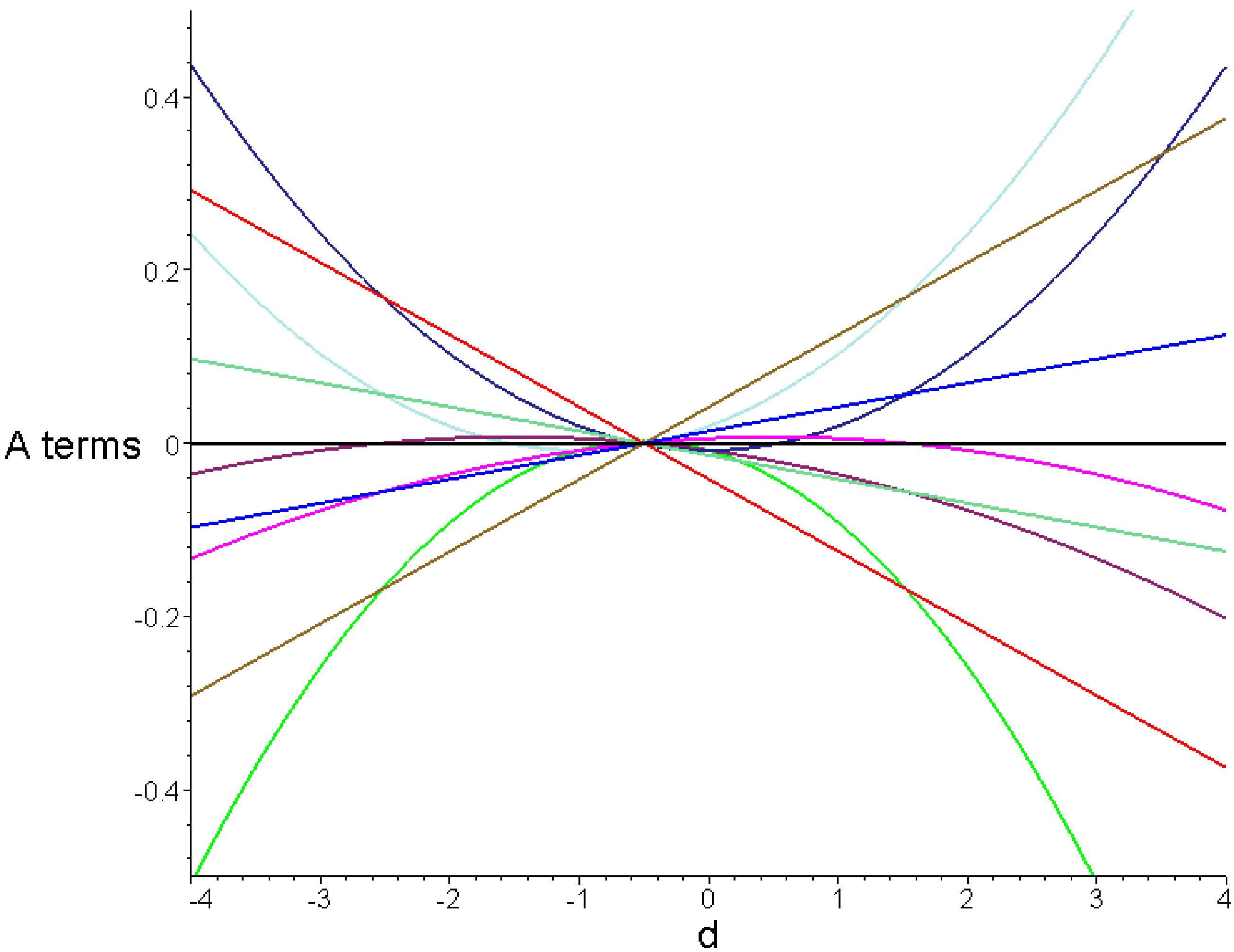
$$
\begin{aligned}
[\,\mathrm{sqrt}(&864c^3d^3 - 72dc^2 - 3 - 432c^2d^4 + 1296c^4d^2 + 216c^2d^2 + 72cd - 864c^3d^2 - 144cd^2 \\
&+ 1296c^4d^4 - 2592d^3c^4 + 432c^2d^3) = Z1\,]
\end{aligned}
$$

$C(\varepsilon_4) = 67532$ *multiplications* $+\ 14939$ *additions* $+\ 1372$ *divisions* $+\ 1298$ *functions*

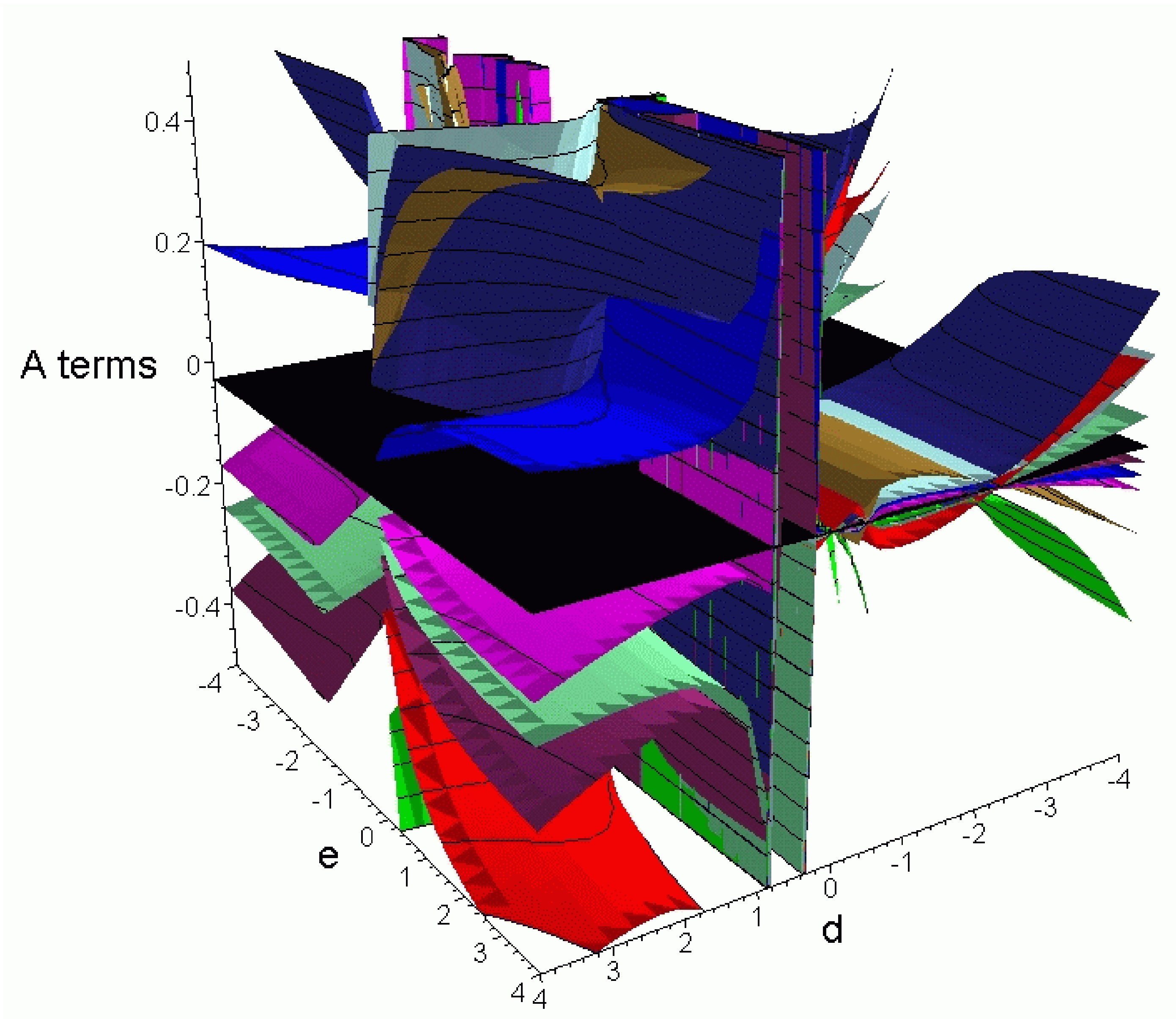
$C(\varepsilon_4) = 4849$ *additions* $+\ 45$ *divisions* $+\ 145$ *functions* $+\ 50360$ *multiplications*

size of error term

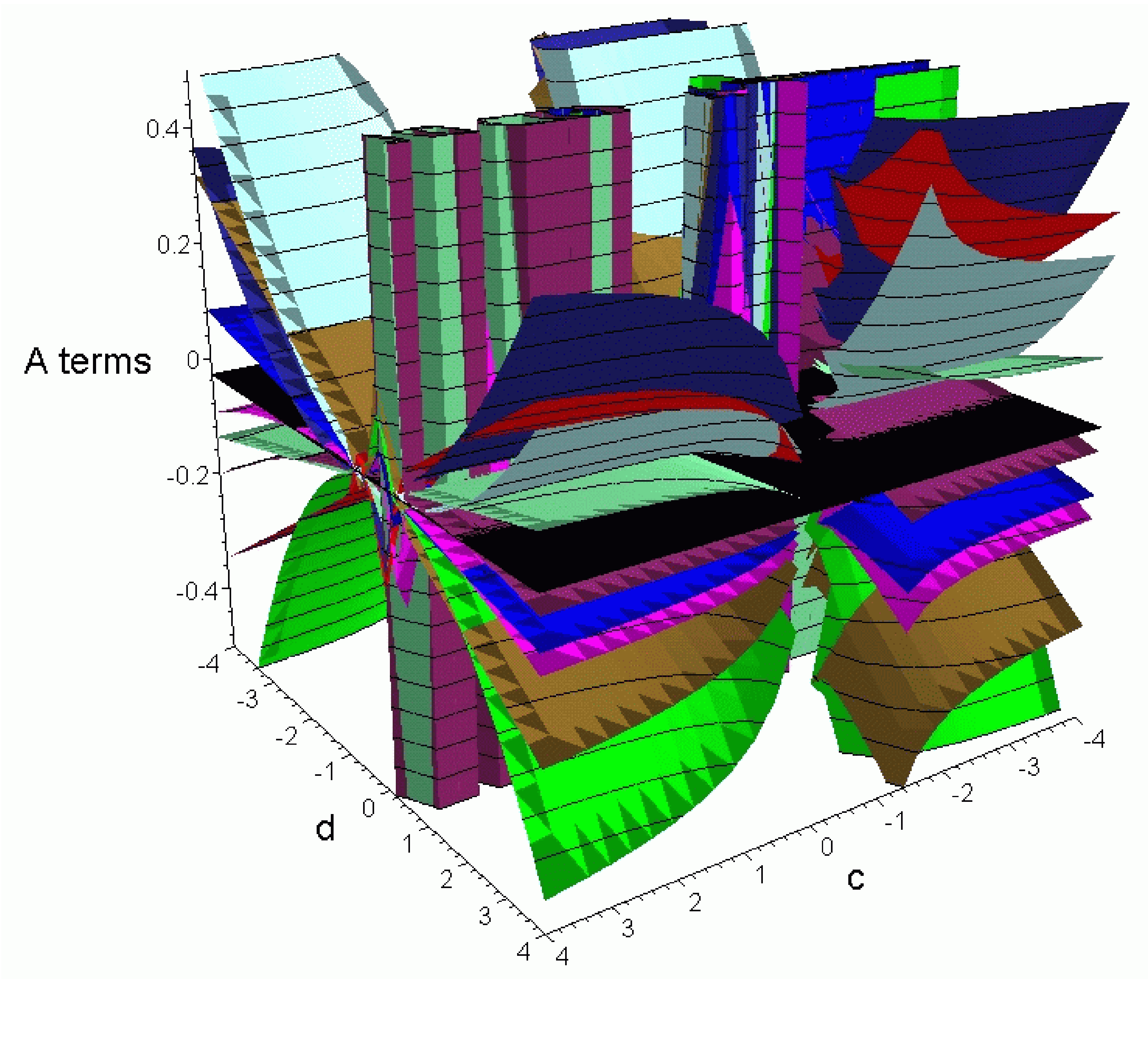
An example of linear-A terms to eliminate



An example of linear-A terms to eliminate



An example of linear-A terms to eliminate



Example of a [7,3] optimized solution

$$S_1 = \left[a = \frac{5}{24}, b = \frac{1}{2}, c = \frac{2}{3}, d = \frac{-1}{2}, e = \frac{-1}{24}, f = 1, g = \frac{1}{6} \right]$$

asymmetric solution

$$\begin{aligned}
\varepsilon_1 = \Big(& -\frac{1}{3456}\,\&^*(A,A,A,B) + \frac{1}{1152}\,\&^*(A,A,B,A) - \frac{1}{288}\,\&^*(A,A,B,B) - \frac{1}{1152}\,\&^*(A,B,A,A) \\
& + \frac{1}{144}\,\&^*(A,B,A,B) + \frac{1}{3456}\,\&^*(B,A,A,A) - \frac{1}{144}\,\&^*(B,A,B,A) + \frac{1}{288}\,\&^*(B,B,A,A) \Big)\tau^4 + \Big(\\
& - \frac{37}{23040}\,\&^*(A,A,B,A,A) + \frac{97}{34560}\,\&^*(A,A,B,A,B) + \frac{47}{34560}\,\&^*(A,A,A,B,A) \\
& - \frac{293}{207360}\,\&^*(A,A,A,B,B) - \frac{19}{46080}\,\&^*(A,A,A,A,B) - \frac{47}{23040}\,\&^*(A,A,B,B,A) \\
& - \frac{1}{320}\,\&^*(A,A,B,B,B) + \frac{1}{1080}\,\&^*(B,A,A,B,A) - \frac{1}{720}\,\&^*(B,A,A,B,B) - \frac{17}{138240}\,\&^*(B,A,A,A,A) \\
& - \frac{17}{51840}\,\&^*(B,A,A,A,B) + \frac{1}{480}\,\&^*(A,B,B,B,A) + \frac{1}{2880}\,\&^*(A,B,B,B,B) + \frac{11}{7680}\,\&^*(A,B,B,A,A) \\
& - \frac{1}{320}\,\&^*(A,B,B,A,B) + \frac{7}{5760}\,\&^*(A,B,A,B,A) + \frac{7}{960}\,\&^*(A,B,A,B,B) + \frac{1}{1280}\,\&^*(A,B,A,A,A) \\
& + \frac{1}{17280}\,\&^*(A,B,A,A,B) + \frac{1}{2880}\,\&^*(B,B,B,B,A) + \frac{1}{2880}\,\&^*(B,B,B,A,A) - \frac{1}{720}\,\&^*(B,B,B,A,B) \\
& + \frac{487}{207360}\,\&^*(B,B,A,A,A) + \frac{1}{480}\,\&^*(B,B,A,A,B) + \frac{1}{2880}\,\&^*(B,B,A,B,A) + \frac{1}{480}\,\&^*(B,B,A,B,B) \\
& - \frac{1}{320}\,\&^*(B,A,B,B,A) - \frac{1}{720}\,\&^*(B,A,B,B,B) - \frac{173}{34560}\,\&^*(B,A,B,A,A) - \frac{1}{720}\,\&^*(B,A,B,A,B) \Big)\tau^5
\end{aligned}$$

linear-A subterms are eliminated from the τ^4 term

Work Progress Report

- SYMPLECTIC is up and running, producing useful results
- Accessible [N,n] parameter space being explored
  - N = number of exponential terms $S_{A,B}(\alpha_k\tau)$
  - n = time step order of method, $O(\tau^n)$
  - symbolic algebra progress thus far:

n\N	2	3	4	5	6	7	8	9
1	-	-	-	-	-	-	-	-
2	×	✓	✓	✓	✓	✓		
3	×	×	×	×	✓	(✓)		
4	×	×	×	×	×	✓	(✓)	
5	×	×	×	×	×	×	×	×

    - Red shaded region is probably beyond current hardware and CAS capabilities
    - Each [N,n] case can yield *many* different solutions

- Optimized methods being compared with numerical solar system tests
- First AJ paper (of two) has been submitted

Preliminary Numerical Results

- Ran two 10,000-year cases:

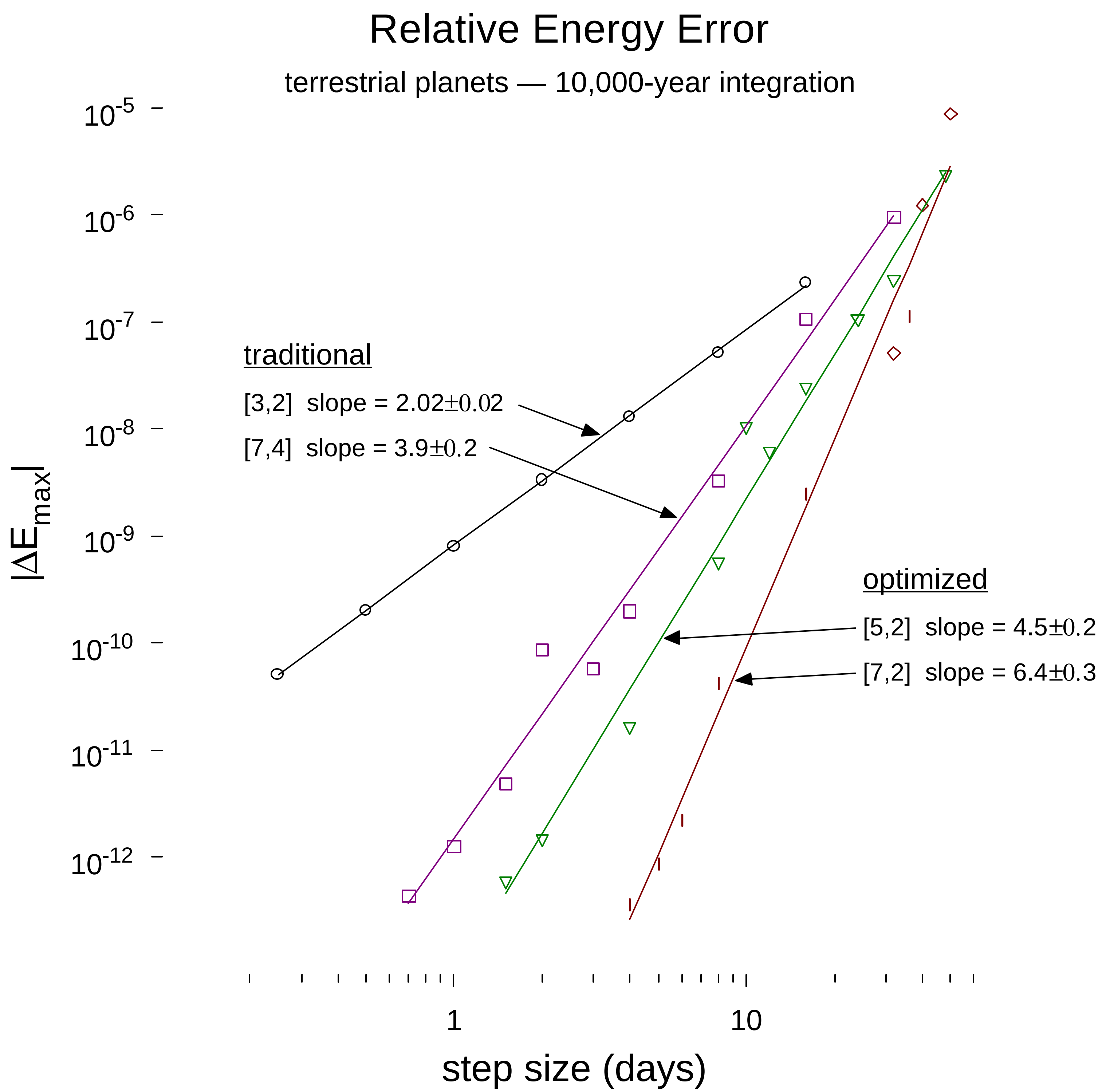
    1. terrestrial planets only ($\varepsilon \sim 10^{-5}$)
    2. all 9 planets ($\varepsilon \sim 10^{-3}$)

- three diagnostic parameters:
    - step size h
    - max energy error
    - elapsed CPU time

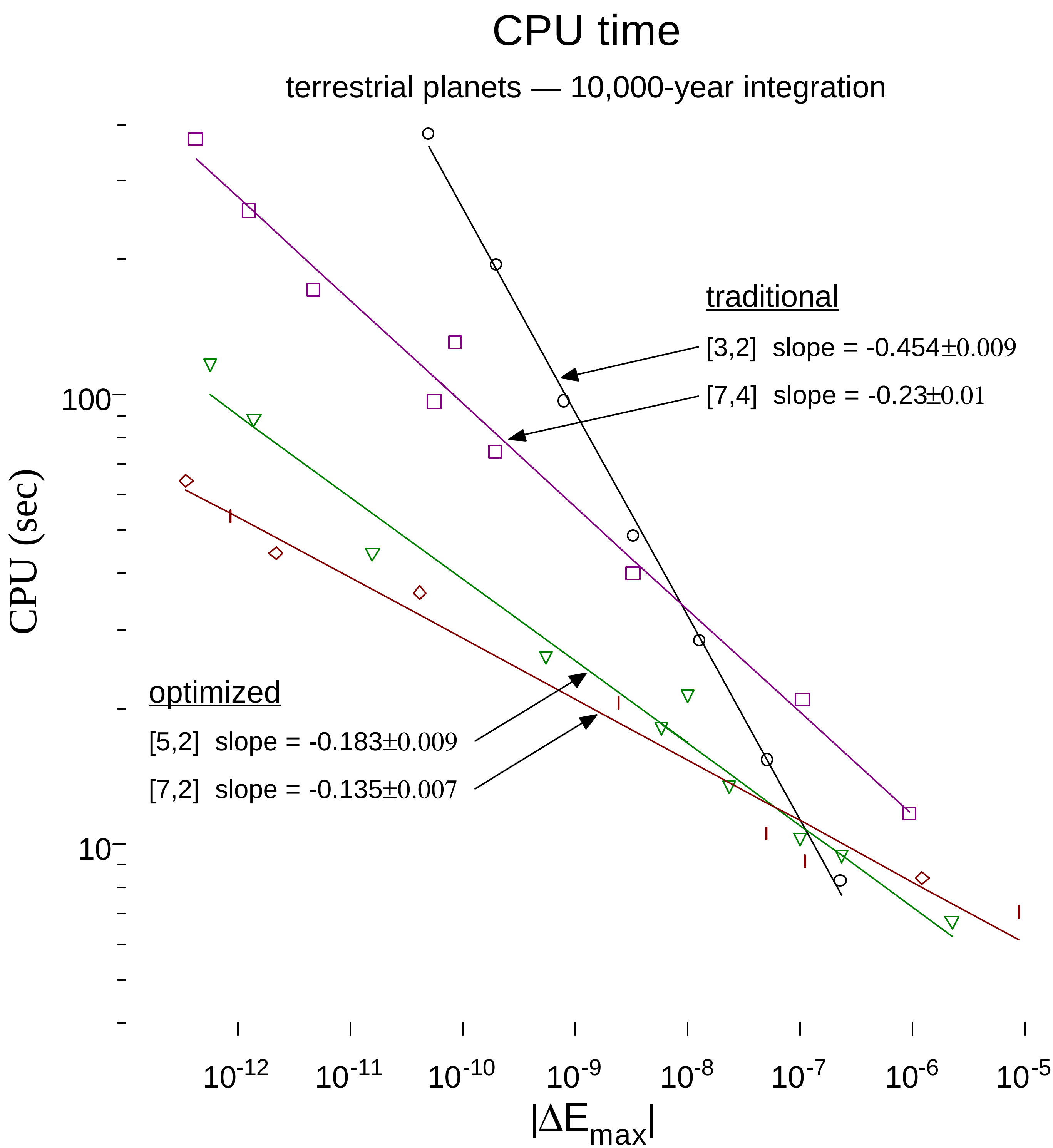
- Compared two selected methods (of many), from among the optimized [5,2] and [7,2] solutions, with the traditional 2nd and 4th order methods [3,2] and [7,4]

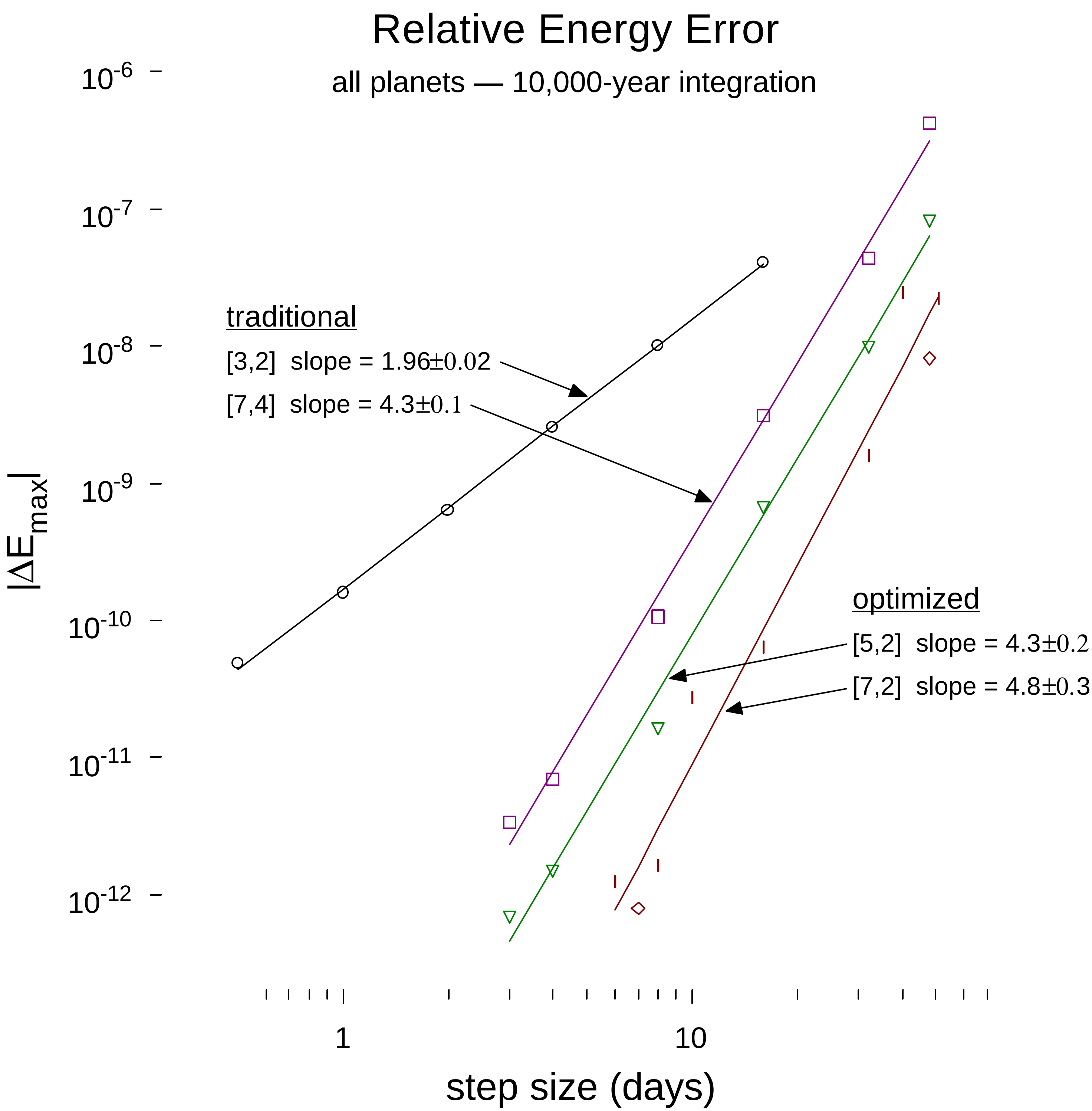
Preliminary Numerical Results (continued)

- Results:
  - For traditional [3,2] and [7,4] methods, max energy error goes as τ^2 and τ^4, as expected
  - For optimized [5,2] and [7,2] methods, max energy error goes as τ^4 and τ^6
  - The optimized methods cost significantly less in CPU time than the traditional methods
    - even [7,2] is less costly than [3,2] at higher accuracies!

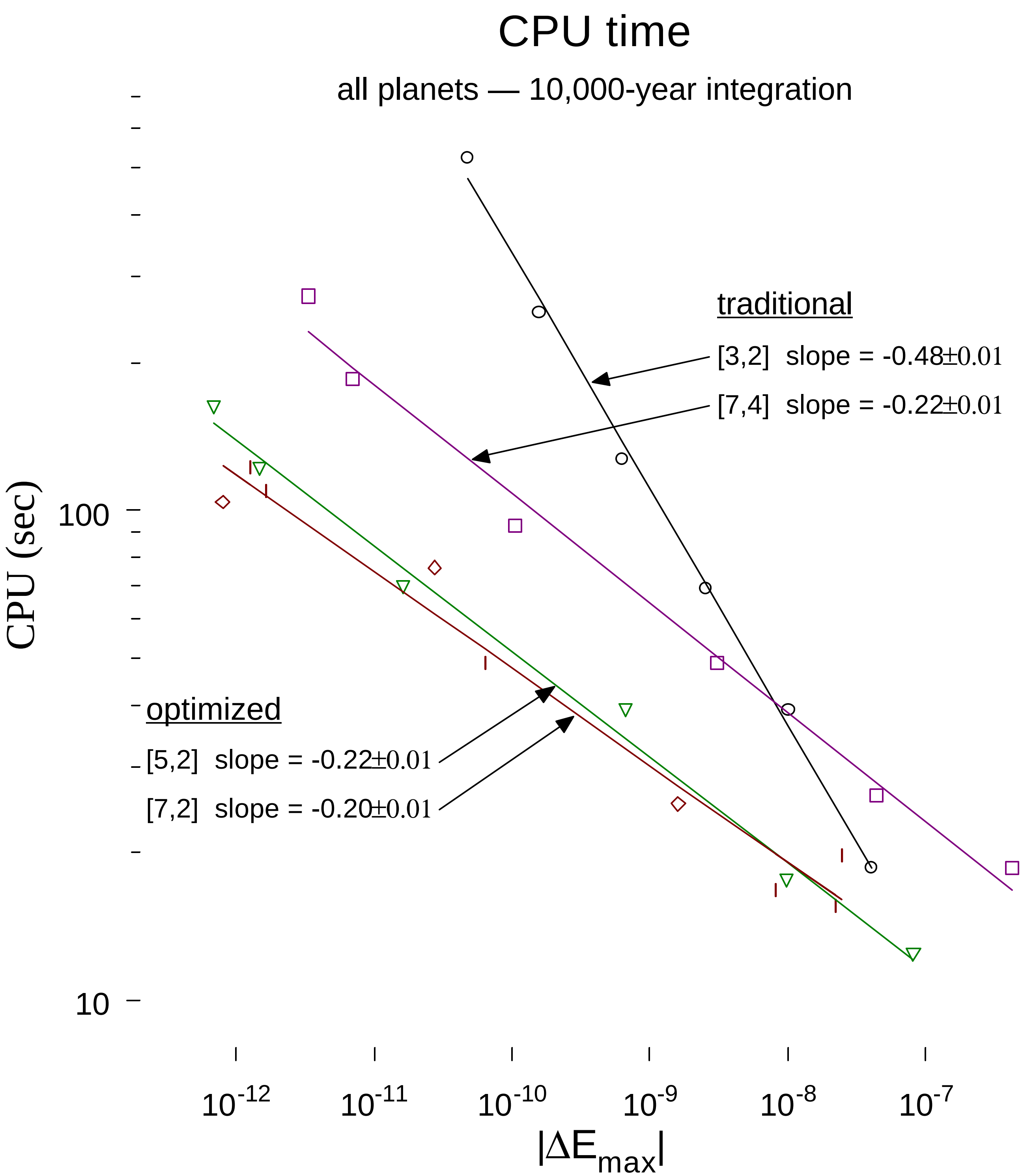


Relative Energy Error
terrestrial planets — 10,000-year integration
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
10^{-11}
10^{-12}
$|\Delta E_{max}|$
traditional
[3,2] slope = 2.02±0.02
[7,4] slope = 3.9±0.2
optimized
[5,2] slope = 4.5±0.2
[7,2] slope = 6.4±0.3
1
10
step size (days)


CPU time
terrestrial planets — 10,000-year integration
CPU (sec)
100
10
traditional
[3,2] slope = -0.454±0.009
[7,4] slope = -0.23±0.01
optimized
[5,2] slope = -0.183±0.009
[7,2] slope = -0.135±0.007
10^{-12}
10^{-11}
10^{-10}
10^{-9}
10^{-8}
10^{-7}
10^{-6}
10^{-5}
$|\Delta E_{max}|$



Relative Energy Error
all planets — 10,000-year integration
$|\Delta E_{max}|$
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
10^{-11}
10^{-12}
traditional
[3,2] slope = 1.96±0.02
[7,4] slope = 4.3±0.1
optimized
[5,2] slope = 4.3±0.2
[7,2] slope = 4.8±0.3
1
10
step size (days)



CPU time
all planets — 10,000-year integration
traditional
[3,2] slope = -0.48±0.01
[7,4] slope = -0.22±0.01
optimized
[5,2] slope = -0.22±0.01
[7,2] slope = -0.20±0.01
CPU (sec)
100
10
10-12
10-11
10-10
10-9
10-8
10-7
|ΔEmax|

Preliminary Conclusions

- This is fun!
- The approach outlined in this talk yields optimized low-order (in time step) symplectic methods that can perform as well as higher-order methods, but at significantly less cost.

To Do

- Teach SYMPLECTIC new tricks
  - better intermediate expression simplification
    - expressions occupy tens of megabytes
    - requires ~100 MB and more of "running room" (i.e., RAM)
    - ➡ handle complicated nested sqrts (easy)
    - ➡ represent error expressions in a commutator notation (hard!)
    - ➡ Take advantage of BCH formula
  - eliminate selected ε^2 subterms
- Complete the exploration of the
  [exp terms, step order]
  space out to current software (Maple) and hardware (memory, speed) limits
- Complete the numerical comparisons of each of the optimized methods
- 2nd AJ paper